Claims

1. A method for implementing a breakpoint debugging scheme comprising:

storing application code in a first memory;

storing a breakpoint service routine in a separate second memory;

encountering a breakpoint request;

switching the active memory from said first memory to said second memory; and

executing said breakpoint service routine stored in said second memory.

2. The method of claim 1 further comprising:

switching the active memory from said second memory to said first memory upon completion of said breakpoint service routine.

3. The method of claim 2 further comprising:

storing status information in a memory stack prior to said switching the active memory from said first memory to said second memory; and

reading said status information from the memory stack upon completion of said breakpoint service routine.

4. The method of claim 1 wherein said second memory is not accessible when said first memory is active.

5. The method of claim 4 further wherein said first memory is not accessible when said second memory is active.

6. A method for implementing a memory scheme comprising:

receiving a breakpoint request;

storing status information in a memory stack prior to said switching the active memory from a first memory to a second memory;

changing memory references from said first memory to said second memory; and

operating a program stored in said second memory.

7. The method of claim 6 further comprising:

restoring memory references from said second memory to said first memory; and

reading said status information from the memory stack upon completion of said breakpoint service routine.

8. The method of claim 6 wherein

said second memory contains a breakpoint service routine.

9. The method of claim 6 further comprising:

resuming operation of a program stored in said first memory.

10. A system for non-intrusive dynamic memory mapping, said system comprising:

a processor;

a first memory for storage of application code;

a second memory for storage of a breakpoint service routine, said second memory configured separate from said first memory; and

a controller coupled between said processor, said first memory, and said second memory, wherein,

said controller is configured to select one of said first memory and said second memory based on the status of said processor.

11. The system of claim 10 wherein said status of said processor includes information regarding the breakpoint status of said processor.

12. The system of claim 10 further comprising a bus interface coupled to said first memory, said second memory, said processor, and said controller.

13. The system of claim 12 wherein said bus interface is configured to send and receive data to and from said first memory and said second memory; and

further wherein said bus interface is further configured to send and receive data to and from said processor based on instructions received from said controller.

14. The system of claim 12 wherein said bus interface is configured to access only one of said first memory and said second memory based on signals provided by said controller.

15. The system of claim 10 wherein said controller is configured to select said second memory when a breakpoint interrupt service request is made.

16. The system of claim 10 wherein said second memory is a read-only memory.